UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/735,433	12/12/2003	Sudarshan Palliyll	JP920030163US1	2308

39903 7590 02/07/2008
IBM ENDICOTT (ANTHONY ENGLAND)
LAW OFFICE OF ANTHONY ENGLAND
PO Box 5307
AUSTIN, TX 78763-5307

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	**PAPER NUMBER**
2168	

MAIL DATE	DELIVERY MODE
02/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/735,433	PALLIYLL ET AL.	
	Examiner	Art Unit	
	Mahesh H. Dwivedi	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,6,7,33,35-37 and 40-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4,6,7,33,35-37 and 40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/4/2007*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. Receipt of Applicant's Amendment, filed on 12/20/2007, is acknowledged. The amendment includes the cancellation of claims 3-5, 8-18, 30-32, 34, 38, and 42, the withdrawal of claims 19-29, and the amending of claims 1, 6, 33, 35-36, and 40.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 12/04/2007 has been received, entered into the record, and considered. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Rejections - 35 USC § 112

3. The rejections in the office action mailed on 09/20/2007 have been overcome by Applicant's amendments received on 12/20/2007.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 1-2, 6-7, 33, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Vermeulen** (U.S. PGPUB 2001/0042171) in view of **Fanning et al.** (U.S. Patent 6,742,023), and in view of **Chapweske** (U.S. PGPUB 2004/0172476).

6. Regarding claim 1, **Vermeulen** teaches a method comprising:

A) computing a set of hash values representing a set of resources stored in association with at least one data processing system within the network (Paragraphs 20 and 24);

B) storing the computed set of hash values (Paragraphs 24 and 32, Figure 5);

C) in response to a requirement for access to a first resource which is accessible via a bandwidth-sensitive connection, retrieving a hash value derived from the required first resource (Paragraph 24);

D) comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values (Paragraph 24);

E) in response to identifying a match for the retrieved hash value (Paragraph 24);

G) wherein the required first resource has bits arranged in a sequence (Paragraph 21);

The examiner notes that **Vermeulen** teaches **"computing a set of hash values representing a set of resources stored in association with at least one data processing system within the network"** as "a basic idea of this invention is to compute a hash code from a file via a given algorithm and to use this hash code to check whether a file to be loaded is already contained in the cache or not" (Paragraph 20) and "Server 14 then computes the hash code of this file 23" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"storing the computed set of hash values"** as "Directory 52 contains a list of the hash codes of the stored files" (Paragraph 32). The examiner further notes that **Vermeulen** teaches **"in response to a requirement for access to a first resource which is accessible via a bandwidth-sensitive connection, retrieving a hash value derived from the required first resource"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It

therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values"** as "Proxy server 12 compares the hashes, 25, to determine whether the requested file is contained in the cache memory or not" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"in response to identifying a match for the retrieved hash value"** as "If the file is already in the cache, it will be immediately transferred, 28, to the client" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"wherein the required first resource has bits arranged in a sequence"** as "Such a hash code is a code word of fixed length generated from a bit sequence of arbitrary length, the word length being determined by the algorithm used. The algorithm is designed so that it is highly unlikely that two different bit sequences will generate the same hash code. Each bit sequence is constituted by an entire file. The length of the hash code may be 128 bits, for example" (Paragraph 21).

Vermeulen does not explicitly teach:

E) initiating retrieval of the required first resource from said at least one data processing system and said bandwidth-sensitive connection;

F) including initiating retrieval of the required first resource via said bandwidth-sensitive connection in parallel with initiating retrieval of the required first resource from said at least one data processing system;

I) and wherein the method includes: combining portions of the bit sequence of said required first resource received via the bandwidth-sensitive connection and received from said at least one data processing system to build the bit sequence of said required first resource; and

J) presenting an indication of the retrieval of the first required resource to a user.

Fanning, however, teaches **"initiating retrieval of the required first resource from said at least one data processing system and said bandwidth-sensitive connection"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identically is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identically may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), **"including initiating retrieval of the required first resource via said bandwidth-sensitive connection in parallel with initiating retrieval of the required first resource from said at least one data processing system"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer

servers 1004, 100 must have the identical file. In one embodiment, file identically is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identically may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), **"and wherein the method includes: combining portions of the bit sequence of said required first resource received via the bandwidth-sensitive connection and received from said at least one data processing system to build the bit sequence of said required first resource"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), and **"presenting an indication of the retrieval <u>of the first required resource</u> to a user"** as "The user interface 118 displays the status of each file transfer as it occurs, along with the estimated time until completion, the filename, the percentage of the file transferred, and an identification string for each distribution application 212 currently connected to the user's file transfer server 120. This information is displayed for both the file transfer server 220 (listing other file transfer clients 214 downloading files from the user's file transfer server 120) and the file transfer client 114 (listing the user's file

transfer client's 120. downloads of files from other file transfer servers 220)" (Column 11, lines 39-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Fanning's** would have allowed **Vermeulen's** to provide a method for allowing high-bandwidth file transfer client to rapidly download parts of the same file from several lower bandwidth file transfer servers, as noted by **Fanning** (Column 8, lines 18-20).

Vermeulen and **Fanning** do not explicitly teach:

H) the step of initiating retrieval of the required first resource from said at least one data processing system and via said bandwidth-sensitive connection comprises: initiating retrieval of all bits of said required first resource from said at least one data processing system and initiating retrieval of all bits of said required resource via said bandwidth-sensitive connection, wherein the retrieval of bits of said required first resource from said at least one data processing system is in a reverse bit sequence order relative to the retrieval of bits of said required first resource via the bandwidth-sensitive connection.

Chapweske, however, teaches **"the step of initiating retrieval of the required first resource from said at least one data processing system and via said bandwidth-sensitive connection comprises: initiating retrieval of all bits of said required first resource from said at least one data processing system and initiating retrieval of all bits of said required resource via said bandwidth-sensitive connection, wherein the retrieval of bits of said required first resource from said at least one data processing system is in a reverse bit sequence order relative to the retrieval of bits of said required first resource via the bandwidth-sensitive connection"** as "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for

any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85), "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97), and "Normally the download is started from the front of the unscheduled ranges, but if the source is too slow to complete the minimum required bytes within the cycle period, then it is required to download from the end of the range. During each new schedule, we check to see which source is the bottleneck in providing the next high priority bytes, if this host has not provided its content within a reasonable time period of the end of the cycle, then the new source is scheduled with the undownloaded bytes of the bottleneck source as the parent for the new download. The advanced scheduler will then schedule the suffix of that range to be downloaded" (Paragraph 101).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Chapweske's** would have allowed **Vermeulen's** and **Fanning's** to provide a method for increasing the speed of downloaded content, as noted by **Chapweske** (Paragraph 97).

Regarding claim 2, **Vermeulen** further teaches a method comprising:

A) wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer via the bandwidth-sensitive connection (Paragraph 24); and

B) receiving the hash value from the server computer via the bandwidth-sensitive connection (Paragraphs 22-24, Figure 1).

The examiner notes that **Vermeulen** teaches **"wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer via the bandwidth-sensitive**

connection" as "If the requested file is not in the cache, proxy server 12 will send a "send file" request, 26, to remote server 14, which then transfers the file, 27, to the proxy server, which stores it in its cache memory and transfers it, 28, to client 11" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"receiving the hash value from the server computer via the bandwidth-sensitive connection"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24).

Regarding claim 6, **Vermeulen** teaches a method comprising:

A) computing a set of hash values representing a set of resources distributed across a plurality of data processing systems within a local area network (LAN), the resources within said set of resources being accessible from respective ones of the plurality of data processing systems (Paragraphs 20 and 24);

B) storing the set of hash values together with an identification of a respective data processing system of said plurality of data processing systems storing the resource corresponding to each of the set of hash values (Paragraph 32);

C) in response to a requirement for access to a required resource which is stored at a remote data processing system, retrieving from the remote data processing system a hash value derived from the required resource (Paragraph 24);

D) comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values (Paragraph 24);

E) in response to identifying a match for the retrieved hash value (Paragraph 24);

G) wherein the required first resource has bits arranged in a sequence (Paragraph 21);

The examiner notes that **Vermeulen** teaches **"computing a set of hash values representing a set of resources distributed across a plurality of data processing systems within a local area network (LAN), the resources within said set of resources being accessible from respective ones of the plurality of data processing systems"** as "a basic idea of this invention is to compute a hash code from a file via a given algorithm and to use this hash code to check whether a file to be loaded is already contained in the cache or not" (Paragraph 20) and "Server 14 then computes the hash code of this file 23" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"storing the set of hash values together with an identification of a respective data processing system of said plurality of data processing systems storing the resource corresponding to each of the set of hash values"** as "Directory 52 contains a list of the hash codes of the stored files and, for each hash code, the memory address at which the associated file is stored in the second memory area 53" (Paragraph 32). The examiner further notes that **Vermeulen** teaches **"in response to a requirement for access to a required resource which is stored at a remote data processing system, retrieving from the remote data processing system a hash value derived from the required resource"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values"** as "Proxy server 12 compares the hashes, 25, to determine whether the requested file is contained in the cache memory or not" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"in response to identifying a match for the retrieved hash value"** as "If the file is already in the

cache, it will be immediately transferred, 28, to the client" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"wherein the required first resource has bits arranged in a sequence"** as "Such a hash code is a code word of fixed length generated from a bit sequence of arbitrary length, the word length being determined by the algorithm used. The algorithm is designed so that it is highly unlikely that two different bit sequences will generate the same hash code. Each bit sequence is constituted by an entire file. The length of the hash code may be 128 bits, for example" (Paragraph 21).

Vermeulen does not explicitly teach:

E) initiating retrieval of the required resource from a respective one of the plurality of data processing systems at which the resource corresponding to the matched hash value is stored and from said remote data processing system;

F) including initiating retrieval of the required resource in parallel with initiating retrieval of the required resource from one of the plurality of data processing systems at which the resource corresponding to the matched hash value is stored;

I) and wherein the method includes: combining portions of the bit sequence of said required resource received via the bandwidth-sensitive connection and received from said at least one data processing system to build the bit sequence of said required resource; and

J) presenting an indication of the retrieval of the required resource to a user.

Fanning, however, teaches **"initiating retrieval of the required resource from a respective one of the plurality of data processing systems at which the resource corresponding to the matched hash value is stored and from said remote data processing system"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer

client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identically is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identically may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), **"including initiating retrieval of the required resource in parallel with initiating retrieval of the required resource from one of the plurality of data processing systems at which the resource corresponding to the matched hash value is stored"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identically is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identically may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), **"and wherein the method includes: combining portions of the bit sequence of said required resource received via the bandwidth-sensitive connection and received from said at least one data processing system to build the bit sequence of said required resource"** as "In an

alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5), and **"presenting an indication of the retrieval of the required resource to a user"** as "The user interface 118 displays the status of each file transfer as it occurs, along with the estimated time until completion, the filename, the percentage of the file transferred, and an identification string for each distribution application 212 currently connected to the user's file transfer server 120. This information is displayed for both the file transfer server 220 (listing other file transfer clients 214 downloading files from the user's file transfer server 120) and the file transfer client 114 (listing the user's file transfer client's 120. downloads of files from other file transfer servers 220)" (Column 11, lines 39-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Fanning's** would have allowed **Vermeulen's** to provide a method for allowing high-bandwidth file transfer client to rapidly download parts of the same file from several lower bandwidth file transfer servers, as noted by **Fanning** (Column 8, lines 18-20).

Vermeulen and **Fanning** do not explicitly teach:

H) the step of initiating retrieval of the required resource from said at least one data processing system comprises: initiating retrieval of the bit sequence of said required resource in a reverse order relative to the retrieval of said required resource via the bandwidth-sensitive connection.

Chapweske, however, teaches **"the step of initiating retrieval of the required resource from said at least one data processing system comprises: initiating retrieval of the bit sequence of said required resource in a reverse order relative to the retrieval of said required resource via the bandwidth-sensitive connection"** as "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85), "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97), and "Normally the download is started from the front of the unscheduled ranges, but if the source is too slow to complete the minimum required bytes within the cycle period, then it is required to download from the end of the range. During each new schedule, we check to see which source is the bottleneck in providing the next high priority bytes, if this host has not provided its content within a reasonable time period of the end of the cycle, then the new source is scheduled with the undownloaded bytes of the bottleneck source as the parent for the new download. The advanced scheduler will then schedule the suffix of that range to be downloaded" (Paragraph 101).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Chapweske's would have allowed **Vermeulen's** and **Fanning's** to provide a method for increasing the speed of downloaded content, as noted by **Chapweske** (Paragraph 97).

Regarding claim 7, **Vermeulen** further teaches a method comprising:
A) wherein the set of hash values and identification of a respective data processing system are stored with information regarding the location within storage of the respective data processing system of the resource corresponding to the hash value (Paragraph 32).

The examiner notes that **Vermeulen** teaches **"wherein the set of hash values and identification of a respective data processing system are stored with information regarding the location within storage of the respective data processing system of the resource corresponding to the hash value"** as "Directory 52 contains a list of the hash codes of the stored files and, for each hash code, the memory address at which the associated file is stored in the second memory area 53" (Paragraph 32).

Regarding claim 33, **Vermeulen** does not explicitly teach a method comprising:
A) retrieving information indication size of the first required resource; and
B) completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource.

Fanning, however, teaches **"retrieving information indication size of the first required resource"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer

servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5) and **"completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Fanning's** would have allowed **Vermeulen's** to provide a method for allowing high-bandwidth file transfer client to rapidly download parts of the same file from several lower bandwidth file transfer servers, as noted by **Fanning** (Column 8, lines 18-20).

Regarding claim 35, **Vermeulen** does not explicitly teach a method comprising:
A) retrieving information indication size of the required resource for use in controlling the combining; and
B) completing the combing responsive to a total number of bits retrieved reaching the indicated size of the required resource.

Fanning, however, teaches **"retrieving information indication size of the required resource for use in controlling the combining"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5) and **"completing the combing responsive to a total number of bits retrieved reaching the indicated size of the required resource"** as "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is

instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Fanning's** would have allowed **Vermeulen's** to provide a method for allowing high-bandwidth file transfer client to rapidly download parts of the same file from several lower bandwidth file transfer servers, as noted by **Fanning** (Column 8, lines 18-20).

7. Claims 36-37, 39-41, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Vermeulen** (U.S. PGPUB 2001/0042171) in view of **Rodriquez** (U.S. PGPUB 2005/0090283), and in view of **Chapweske** (U.S. PGPUB 2004/0172476).

8. Regarding claim 36, **Vermeulen** teaches a computer program product comprising:

A) storing a set of hash values representing a set of resources, the resources being stored within the LAN (Paragraphs 24 and 32, Figure 5);

B) in response to a requirement for access to a required first resource which is accessible via the remote network, retrieving a hash value derived from the required first resource (Paragraph 24);

C) comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values (Paragraph 24);

D) in response to identifying a match for the retrieved hash value (Paragraph 24);

G) wherein the required first resource has bits arranged in a sequence (Paragraph 21)

The examiner notes that **Vermeulen** teaches **"storing a set of hash values representing a set of resources, the resources being stored within the LAN"** as "Directory 52 contains a list of the hash codes of the stored files" (Paragraph 32). The examiner further notes that **Vermeulen** teaches **"in response to a requirement for access to a required first resource which is accessible via the remote network, retrieving a hash value derived from the required first resource"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values"** as "Proxy server 12 compares the hashes, 25, to determine whether the requested file is contained in the cache memory or not" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"in response to identifying a match for the retrieved hash value"** as "If the file is already in the cache, it will be immediately transferred, 28, to the client" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"wherein the required first resource has bits arranged in a sequence"** as "Such a hash code is a code word of fixed length generated from a bit sequence of arbitrary length, the word length being determined by the algorithm used. The algorithm is designed so that it is highly unlikely that two different bit sequences will generate the same hash code. Each bit sequence is constituted by an entire file. The length of the hash code may be 128 bits, for example" (Paragraph 21).

Vermeulen does not explicitly teach:

E) initiating retrieval of the required first resource from said LAN and from said remote network;

F) including initiating retrieval of the required first resource from said LAN in parallel with initiating retrieval of the required first resource from said remote network;
I) wherein the instructions when executed by the computer, cause the computer to implement the method such that the method further comprises the steps of: combining a portion of the bit sequence of said required first resource received from the LAN and a portion of the bit sequence of said required first resource from the remote network to build the bit sequence of said required first resource.

Rodriquez, however, teaches **"initiating retrieval of the required first resource from said at least one data processing system and said bandwidth-sensitive connection"** as "Wireless network access device further comprises at least one local network interface, and in the disclosed embodiment comprises multiple local network interfaces 170, 172. In an exemplary embodiment, wireless network access device 130 operates one or more local communication networks, such as a local area network (LAN). Local network interface 170 may be embodied as a wireless PCMCIA card that provides access to a wireless data network using one of a plurality of wireless networking protocols, e.g., Bluetooth, or 802.11(b). Local network interface 172 may be embodied as a PCMCIA card that provides access to a wired network, e.g., a LAN. Software for operating the local communication network may reside on the PCMCIA cards, or as one of the application programs 160 that execute on the processing unit 132 of wireless network access device 130" (Paragraph 26) and "In an exemplary embodiment, wireless access device 130 is configured to maximize the available bandwidth to users. Accordingly, wireless access device 130 activates all available wireless network interfaces to establish a plurality of wireless communication connections that may be spread among different communication service providers. At operation 320, wireless network access device 130 transmits the request for the resource from the activated communication interface(s). If a plurality of interfaces were activated, then the resource request may be divided among the plurality of interfaces. For example, if a requested web page includes five objects, each of which requires a separate TCP connection, and there are five available communication interfaces, then

each available wireless network interface may be assigned to transmit one a TCP connection" (Paragraphs 31-32), **"including initiating retrieval of the required first resource via said bandwidth-sensitive connection in parallel with initiating retrieval of the required first resource from said at least one data processing system"** as "Operations 616-618 avoid downloading very large objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of images that is ten times the size of the next largest object. The scheduler may break the series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50), and **"wherein the instructions when executed by the computer, cause the computer to implement the method such that the method further comprises the steps of: combining a portion of the bit sequence of said required first resource received from the LAN and a portion of the bit sequence of said required first resource from the remote network to build the bit sequence of said required first resource"** as "Operations 616-618 avoid downloading very large

objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of images that is ten times the size of the next largest object. The scheduler may break the series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rodriquez's** would have allowed **Vermeulen's** to provide a method to avoid downloading large content over single interfaces and/or connection which would result in longer download times, as noted by **Rodriquez** (Paragraph 49).

Vermeulen and **Rodriguez** do not explicitly teach:

J) the step of initiating retrieval of the required first resource from said LAN and from said remote network comprises: initiating retrieval of all bits of said required first resource of said required first resource from the LAN and initiating retrieval of all bits of said required first resource from said remote network, wherein the retrieval of bits of

said required first resource from said LAN is in a reverse bit sequence order relative to the retrieval of bits of said resource from the remote network;

K) presenting an indication of the retrieval of the required first resource to a user.

Chapweske, however, teaches **"the step of initiating retrieval of the required first resource from said LAN and from said remote network comprises: initiating retrieval of all bits of said required first resource of said required first resource from the LAN and initiating retrieval of all bits of said required first resource from said remote network, wherein the retrieval of bits of said required first resource from said LAN is in a reverse bit sequence order relative to the retrieval of bits of said resource from the remote network"** as "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85), "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97), and "Normally the download is started from the front of the unscheduled ranges, but if the source is too slow to complete the minimum required bytes within the cycle period, then it is required to download from the end of the range. During each new schedule, we check to see which source is the bottleneck in providing the next high priority bytes, if this host has not provided its content within a reasonable time period of the end of the cycle, then the new source is scheduled with the undownloaded bytes of the bottleneck source as the parent for the new download. The advanced scheduler will then schedule the suffix of that range to be downloaded" (Paragraph 101), and **"presenting an indication of the retrieval of the required first**

resource to a user" as "The control interface allows external applications or users to control the behavior of the parallel download system. The control interface provides a number of features...The download can be started, stopped, suspended, or resumed" (Paragraphs 40-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Chapweske's** would have allowed **Vermeulen's** and **Rodriguez's** to provide a method for increasing the speed of downloaded content, as noted by **Chapweske** (Paragraph 97).

Regarding claim 37, **Vermeulen** further teaches a computer program product comprising:

A) wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer at the remote network (Paragraph 24); and

B) receiving the hash value from the server computer (Paragraphs 22-24, Figure 1).

The examiner notes that **Vermeulen** teaches **"wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer at the remote network"** as "If the requested file is not in the cache, proxy server 12 will send a "send file" request, 26, to remote server 14, which then transfers the file, 27, to the proxy server, which stores it in its cache memory and transfers it, 28, to client 11" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"receiving the hash value from the server computer"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of

this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24).

Regarding claim 39, **Vermeulen** does not explicitly teach a computer program product comprising:

A) retrieving information indication size of the first required resource; and

B) completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource.

Rodriquez, however, teaches **"retrieving information indication size of the first required resource"** as "In another exemplary implementation, an apparatus is provided. The apparatus comprises at least one local communication network interface for receiving a request for a resource. In addition, the apparatus comprises a plurality of wireless network interfaces for transmitting resource requests over wireless communication connections. The apparatus further comprises a memory module, and a processor that executes logic instructions that configure the processor to terminate the received request, determine a number of available wireless network interfaces, determine a number of objects in the resource and the size of each object, and assign each object to at least one available wireless network interface" (Paragraph 7) and **"completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource"** as "Operations 616-618 avoid downloading very large objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of images that is ten times the size of the next largest object. The scheduler may break the

series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rodriquez's** would have allowed **Vermeulen's** to provide a method to avoid downloading large content over single interfaces and/or connection which would result in longer download times, as noted by **Rodriquez** (Paragraph 49).

Regarding claim 40, **Vermeulen** teaches a computer system comprising:

A) a processor (Paragraphs 24 and 37);

B) a storage device connected to the processor (Paragraph 24);

C) wherein the storage device has stored thereon a program for accessing resources within a data processing network (Paragraph 24);

D) the data processing network including a local area network ("LAN") and a remote network outside the LAN (Paragraph 33); and

E) wherein the processor is operative with the program to execute the program for performing the steps of: storing a set of hash values representing a set of resources, the resources being stored within the LAN (Paragraphs 24 and 32, Figure 5);

F) in response to a requirement for access to a required first resource which is accessible via the remote network, retrieving a hash value derived from the required first resource (Paragraph 24);

G) comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values (Paragraph 24);

I) in response to identifying a match for the retrieved hash value (Paragraph 24);

J) wherein the required first resource has bits arranged in a sequence (Paragraph 21).

The examiner notes that **Vermeulen** teaches "wherein the processor is operative with the program to execute the program for performing the steps of: storing a set of hash values representing a set of resources, the resources being stored within the LAN" as "Directory 52 contains a list of the hash codes of the stored files" (Paragraph 32). The examiner further notes that **Vermeulen** teaches **"in response to a requirement for access to a required first resource which is accessible via the remote network, retrieving a hash value derived from the required first resource"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"comparing the retrieved hash value with the stored set of hash values to identify a match between the retrieved hash value and any of the stored set of hash values"** as "Proxy server 12 compares the hashes, 25, to determine whether the requested file is contained in the cache memory or not" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"in response to identifying a match for the retrieved hash value"** as "If the file is already in the cache, it will be immediately transferred, 28, to the client" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"wherein the required first resource has bits arranged in a sequence"** as "Such a hash code is a code word of fixed length generated from a bit sequence of arbitrary length, the word length being determined by the algorithm used. The algorithm is designed so that

it is highly unlikely that two different bit sequences will generate the same hash code. Each bit sequence is constituted by an entire file. The length of the hash code may be 128 bits, for example" (Paragraph 21).

Vermeulen does not explicitly teach:

H) initiating retrieval of the required first resource from said LAN and remote network;

I) including initiating retrieval of the required first resource from said LAN in parallel with initiating retrieval of the required first resource from said remote network;

L) wherein the instructions when executed by the computer, cause the computer to implement the method such that the method further comprises the steps of: combining a portion of the bit sequence of said required first resource received from the LAN and a portion of the bit sequence of said required first resource from the remote network to build the bit sequence of said required first resource.

Rodriquez, however, teaches **"initiating retrieval of the required first resource from said LAN and remote network"** as "Wireless network access device further comprises at least one local network interface, and in the disclosed embodiment comprises multiple local network interfaces 170, 172. In an exemplary embodiment, wireless network access device 130 operates one or more local communication networks, such as a local area network (LAN). Local network interface 170 may be embodied as a wireless PCMCIA card that provides access to a wireless data network using one of a plurality of wireless networking protocols, e.g., Bluetooth, or 802.11(b). Local network interface 172 may be embodied as a PCMCIA card that provides access to a wired network, e.g., a LAN. Software for operating the local communication network may reside on the PCMCIA cards, or as one of the application programs 160 that execute on the processing unit 132 of wireless network access device 130" (Paragraph 26) and "In an exemplary embodiment, wireless access device 130 is configured to maximize the available bandwidth to users. Accordingly, wireless access device 130 activates all available wireless network interfaces to establish a plurality of wireless communication connections that may be spread among different communication service providers. At operation 320, wireless network access device 130 transmits the request

for the resource from the activated communication interface(s). If a plurality of interfaces were activated, then the resource request may be divided among the plurality of interfaces. For example, if a requested web page includes five objects, each of which requires a separate TCP connection, and there are five available communication interfaces, then each available wireless network interface may be assigned to transmit one a TCP connection" (Paragraphs 31-32) and **"including initiating retrieval of the required first resource from said LAN in parallel with initiating retrieval of the required first resource from said remote network"** as "Operations 616-618 avoid downloading very large objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of images that is ten times the size of the next largest object. The scheduler may break the series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50), and **"wherein the instructions when executed by the computer, cause the computer to implement the method such that the method further comprises the steps of: combining a portion of the**

bit sequence of said required first resource received from the LAN and a portion of the bit sequence of said required first resource from the remote network to build the bit sequence of said required first resource" as "Operations 616-618 avoid downloading very large objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of images that is ten times the size of the next largest object. The scheduler may break the series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rodriquez's** would have allowed **Vermeulen's** to provide a method to avoid downloading large content over single interfaces and/or connection which would result in longer download times, as noted by **Rodriquez** (Paragraph 49).

Vermeulen and **Rodriguez** do not explicitly teach:

K) the step of initiating retrieval of the required first resource from said LAN and from said remote network comprises: initiating retrieval of all bits of said required first resource of said required first resource from the LAN and initiating retrieval of all bits of said required first resource from said remote network, wherein the retrieval of bits of said required first resource from said LAN is in a reverse bit sequence order relative to the retrieval of bits of said resource from the remote network;

M) presenting an indication of the retrieval of the required first resource to a user.

Chapweske, however, teaches **"the step of initiating retrieval of the required first resource from said LAN and from said remote network comprises: initiating retrieval of all bits of said required first resource of said required first resource from the LAN and initiating retrieval of all bits of said required first resource from said remote network, wherein the retrieval of bits of said required first resource from said LAN is in a reverse bit sequence order relative to the retrieval of bits of said resource from the remote network"** as "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85), "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97), and "Normally the download is started from the front of the unscheduled ranges, but if the source is too slow to complete the minimum required bytes within the cycle period, then it is required to download from the end of the range. During each new schedule, we check to see which source is the bottleneck in providing the next high priority bytes, if this host has not provided its content within a reasonable

time period of the end of the cycle, then the new source is scheduled with the undownloaded bytes of the bottleneck source as the parent for the new download. The advanced scheduler will then schedule the suffix of that range to be downloaded" (Paragraph 101), and **"presenting an indication of the retrieval to a user"** as "The control interface allows external applications or users to control the behavior of the parallel download system. The control interface provides a number of features...The download can be started, stopped, suspended, or resumed" (Paragraphs 40-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Chapweske's** would have allowed **Vermeulen's** and **Rodriguez's** to provide a method for increasing the speed of downloaded content, as noted by **Chapweske** (Paragraph 97).

Regarding claim 41, **Vermeulen** further teaches a computer system comprising:
A) wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer at the remote network (Paragraph 24); and
B) receiving the hash value from the server computer (Paragraphs 22-24, Figure 1).

The examiner notes that **Vermeulen** teaches **"wherein the step of retrieving the hash value derived from the required first resource comprises: sending a resource access request to a server computer at the remote network"** as "If the requested file is not in the cache, proxy server 12 will send a "send file" request, 26, to remote server 14, which then transfers the file, 27, to the proxy server, which stores it in its cache memory and transfers it, 28, to client 11" (Paragraph 24). The examiner further notes that **Vermeulen** teaches **"receiving the hash value from the server computer"** as "Client therefore sends a file request 21 with the address of the requested file to proxy server 12. To be able to check its cache as to whether the requested file is contained therein, proxy server 12 needs the hash code belonging to the file. It therefore sends to remote server 14 a hash request 22 in which the

requested file is specified with its address. Server 14 then computes the hash code of this file, 23, and sends a message containing the computed hash code back to the Proxy server" (Paragraph 24).

Regarding claim 43, **Vermeulen** does not explicitly teach a computer system comprising:

A) retrieving information indication size of the first required resource; and

B) completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource.

Rodriquez, however, teaches **"retrieving information indication size of the first required resource"** as "In another exemplary implementation, an apparatus is provided. The apparatus comprises at least one local communication network interface for receiving a request for a resource. In addition, the apparatus comprises a plurality of wireless network interfaces for transmitting resource requests over wireless communication connections. The apparatus further comprises a memory module, and a processor that executes logic instructions that configure the processor to terminate the received request, determine a number of available wireless network interfaces, determine a number of objects in the resource and the size of each object, and assign each object to at least one available wireless network interface" (Paragraph 7) and **"completing the combing responsive to a total number of bits retrieved reaching the indicated size of the first required resource"** as "Operations 616-618 avoid downloading very large objects over a single TCP connection, which is transmitted across a single wireless network interface. Each wireless network interface has limited bandwidth. Therefore, downloading very large objects over a single TCP connection may unnecessarily delay the overall download time. Accordingly, the threshold may be set at a fixed number that is a function of the bandwidth available on a particular wireless network interface. Alternatively, the threshold may be set as a function of the size of a particular object in relation to the size of the other objects in the resource. For example, assume a particular resource includes five objects, one of which is a series of

images that is ten times the size of the next largest object. The scheduler may break the series of images down into a number of individual objects that may be transmitted over separate TCP connections. Setting the threshold in this manner should reduce the total transmission time for the resource. It is possible to determine the threshold based on other factors; the thresholds described herein are merely exemplary" (Paragraph 49) and "Based on the number of objects in the resource and whether one or more objects were subdivided in operation 616, the parallel scheduler module determines a number of TCP connections to use in downloading the resource, at operation 620. For example, assume a resource includes five objects, but one object is subdivided into three separate TCP connections. The parallel scheduler module would assign seven TCP connections to download this resource" (Paragraph 50).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rodriquez's** would have allowed **Vermeulen's** to provide a method to avoid downloading large content over single interfaces and/or connection which would result in longer download times, as noted by **Rodriquez** (Paragraph 49).

Response to Arguments

9. Applicant's arguments filed 12/20/2007 have been fully considered but they are not persuasive.

Applicant argues on pages 12-13 that **"If Chapweske was teaching that retrieval is initiated for an entire file from one source and retrieval is also initiated from the same entire file from another source, but in reverse bit sequence order, then why would Chapweske need terminology to facilitate assigning the last "N-" set of bytes to a new source?"**. However, the examiner wishes to state that the Office action is only concerned with facts and not hypothetical questions. Moreover, **Chapweske** teaches retrieval of bits in reverse order (i.e. starting retrieval from the last bit) (See "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all

normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...0___Inclusive list of all bytes from 0 to infinity, or end of the file. 20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85)). Furthermore, the examiner wishes to refer to Paragraph 97 of **Chapweske** which states it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97). The examiner further wishes to state that **Chapweske** clearly teaches retrieving all bits from the second source because the second source continues to download until it meets up with the first source (see "Once the other source catches up to the preempting source, the download from the other source is canceled "). Moreover, the examiner wishes to state that **Chapweske** teaches that one of his retrieval range is the entire set of bits (see "0___Inclusive list of all bytes from 0 to infinity, or end of the file")

Applicant argues on page 13 that **"That is, since two retrievals...such that the two downloads meet, so to speak, at some non-predetermined location in between the beginning and the end of the file), this is not consistent with teaching by Chapweske about determining in advance a specific size of one of the portions to be downloaded"**. However, the examiner wishes to refer to Paragraph 97 of **Chapweske** which states "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97). The examiner further wishes to state that **Chapweske** clearly teaches two sources meeting up when downloading at opposite ends of the bit sequence.

Applicant argues on page 13 that **"Thus, it should be understood that Chapweske does not teach retrieval in opposing bit sequence order as in the**

present invention, but rather teaches retrieval of an ending portion of a file, which begins at N predetermined bytes from the end of the file...from the end". However, the examiner wishes to refer to Paragraphs 81-85 and 97 of **Chapweske** which state "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...0___Inclusive list of all bytes from 0 to infinity, or end of the file. 20---Negative ranges start from the end of the file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85) and "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97). The examiner further wishes to state that **Chapweske** clearly teaches starting at that last bit (see "Negative ranges start from the end of the file").

Applicant argues on page 14 that **"merely indicates that both the retrievals are initiated for the entire set of bits of the file. This is different than Chapweske, which teaches that at least one of the targeted retrievals for a file is initiated for a predetermined set of bytes...same sequence"**. However, the examiner wishes to refer to Paragraphs 81-85 and 97 of **Chapweske** which state "As an optimization, the prioritization scheduler schedules ranges of bytes to be downloaded. A range is essentially a run length encoding of a sequential list of bytes. A set of bytes can then be compactly represented as a range set which allows all normal mathematical set operations to be efficiently performed on the list of bytes. The invention may utilize range sets extensively in its implementation for any task that requires keeping track of a list of bytes. Examples of ranges and range sets are as follows...0___Inclusive list of all bytes from 0 to infinity, or end of the file. 20---Negative ranges start from the end of the

file. So this range is a list of the last 20 bytes in the file" (Paragraphs 81-85) and "it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled" (Paragraph 97). The examiner further wishes to state that **Chapweske** clearly teaches starting at that last bit (see "Negative ranges start from the end of the file"). Moreover, because **Chapweske** teaches a first source downloading via a range of entire bits (see "0___Inclusive list of all bytes from 0 to infinity, or end of the file") and that a second source continues to download until meeting the first source (see ""it downloads from the end of the undownloaded range, preempting some of the range that has already been scheduled for another source. Once the other source catches up to the preempting source, the download from the other source is canceled"), then as a result, **Chapweske** teaches downloading all bits from opposite ends from two separate sources.

Applicant argues on page 15 that **"Even taken collectively, the references affirmatively teach away from the present invention"**. However, Applicants are also reminded that in order to disqualify a reference based on a "teach away" reasoning, the reference has to *explicitly* suggest or disclose the so-called teach away steps - Applicants assertion can not be accepted if it is unsupported by a valid evidence. In this case, applicants provide no evidence that **Chapweske** teaches away from the invention. The examiner has clearly shown that **Chapweske** teaches parallel downloading of bits of opposite ends of a file.

Applicant argues on page 14 that **"It does not teach or suggest "completing the combining responsive to a total number of bits retrieved reaching the indicated size of the first required resource" as claimed"**. However, the examiner wishes to point to Columns 12-13 of **Fanning** which state "In an alternative embodiment, a parallel download module 1002, as shown in FIG. 8, facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 1004, 1006. The parallel download module 1002 selects the best of at least two file transfer servers 1104, 1006 from among all the file transfer

servers associated with the file descriptions in the search response. Then, the user's file transfer client 1000 is instructed to download different sections of the same file from each of the file transfer servers 1004, 1006. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file transfer load across several different file transfer servers 1004, 1006. Note that for this to function properly, both file transfer servers 1004, 100 must have the identical file. In one embodiment, file identicality is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identicality may be used here, and such methods are well known in the art" (Column 12, lines 53-67-Column 13, lines 1-5). The examiner further wishes to state that it is common knowledge that after downloading different portions of a file simultaneously (i.e., paraloading), reassembly and reordering of the file is accomplished, i.e., after downloading every portion of the file (100% of it) only then is reassembly started. Moreover, one does not download 1.2 MB of a 1.0 MB file; it only downloads 1.0MB of a 1.0MB file.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,883,135 issued to **Obata et al.** on 19 April 2005. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,647,421 issued to **Logue et al.** on 11 November 2003. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,212,521 issued to **Minami et al.** on 19 August 2004. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,098,079 issued to **Howard** on 01 August 2000. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,754,657 issued to **Lomet** on 22 April 2004. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. PGPUB 2004/0068652 issued to **Carpentier et al.** on 08 April 2004. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. PGPUB 2002/0038296 issued to **Margolus et al.** on 28 March 2002. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,434,533 issued to **Sekiguchi et al.** on 13 August 2002. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. PGPUB 2005/0138081 issued to **Alshab et al.** on 23 June 2005. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

Article entitled "Performance Analysis of a Dynamic Parallel Downloading Scheme from Mirror Sites Throughout the Internet" by **Miu et al.**, dated 21 March 2000. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,295,610 issued to **Ganesh et al.** on 25 September 2001. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

U.S. Patent 6,438,606 issued to **Ward** on 20 August 2002. The subject matter disclosed therein is pertinent to that of claims 1-2, 6-7, 33, 35-37, 39-41, and 43 (e.g., methods to quickly obtain requested files remotely via multiple sources).

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner

Art Unit 2168

January 31, 2008

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100